UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/749,924	12/31/2003	Darrell S. McGinnis	INTEL2	6681

6980 7590 07/12/2006

TROUTMAN SANDERS LLP
600 PEACHTREE STREET , NE
ATLANTA, GA 30308

EXAMINER	
IWASHKO, LEV	
ART UNIT	**PAPER NUMBER**
2186	

DATE MAILED: 07/12/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/749,924	**Applicant(s)** MCGINNIS, DARRELL S.	
	Examiner Lev I. Iwashko	**Art Unit** 2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *4/25/2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *31 December 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

SCElmore

STEPHEN C. ELMORE
PRIMARY EXAMINER

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. There were no cancelled or amended claims in the Applicant's arguments.

2. Claims 1-20 stand rejected.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1-20 are rejected under 35 U.S.C. 102(b) as being anticipated by Hagersten et al. (US Patent 5,829,033).

Claim 1. An apparatus, comprising:

- a transaction queue to store pending device transactions and dispatched device transactions, *(Column 30, lines 24-30 – State that there is an input queue that stores transactions)*
- the queue including an input for receiving transactions, the queue including an output for dispatching transactions to a device, *(Column 30, lines 36-52 – Describe the queue as being able to input and output transactions)*
- the device transactions including device read transactions *(Column 13, lines 14 – Claims read transactions)*
- and device write transactions; and *(Column 16, lines 32 – Claims write transactions)*
- a controller coupled to the transaction queue and responsive to an invalid data signal for a device read transaction to prevent the transaction queue from dispatching a pending transaction to the device,

to cause the transaction queue to dispatch again the device read transaction which resulted in the invalid data signal and, subsequently, to send to a desired destination a data available signal for the data which resulted from the device read transaction which was dispatched again, and to enable the transaction queue to dispatch a pending transaction to the device. *(Column 19, lines 4-24 – State the following: "Preferably, the ReadToShareFork can simultaneously (a) write valid data to a requested memory location containing invalid data, and (b) return the valid data to the requestor whose request for the data is still outstanding. Thus, rather than completing the outstanding request in approximately twenty clock cycles, the request is granted within about ten clock cycles at the same time that the data is updated in the relevant requested memory location. Note that from the standpoint of a memory controller the ReadToShareFork appears to implement a write transaction. However, from the standpoint of an address controller the ReadToShareFork appears to be implementing a read transaction")*

Claim 2. The apparatus of claim 1 wherein the controller is further responsive to the invalid data signal to determine if invalid data read from the device location has occurred previously and, if not, to then cause the transaction queue to dispatch again the device read transaction which resulted in the invalid data signal and, subsequently, to send to a desired destination a data available signal for the data which resulted from the device read transaction which was dispatched again, and to enable the transaction queue to dispatch a pending transaction to the device. *(Column 23, lines 51-67 and Column 24, lines 1-3 – State the following: "An asserted IGNORE signal blocks all address controllers from adding an IGNORE-flagged transaction into each address controller's coherent input queue. Thus, not even the address controller associated with a requesting device can load an IGNORE-flagged transaction into its own coherent input*

queue. The IGNORE-flagged transaction does not become asserted upon the omnibus system 20", and thus in a temporal or timing sense, the transaction has not yet occurred. (The transaction will be reissued by the device that asserted the IGNORE signal.) In essence, all address controllers 180 coupled to omnibus system 20" function as though the subject transaction had not occurred. The subject memory address (in which an invalid version of the desired data is retained) may still be accessed, but to no practical purpose since another memory location (which may be anywhere physically) holds the valid version of the desired data. When any CPU or other device attempts to access a cache line associated with an invalid state, as signified by associated memory TAG (or "MTAG"), global network interface 15 will cause the IGNORE signal to be asserted")

Claim 3. The apparatus of claim 1 wherein the controller is further responsive to the invalid data signal to determine if invalid data read from the device location has occurred previously and, if so, then to send to the desired destination a data available signal for the data which resulted from the device read transaction which resulted in the invalid data signal. *(Column 19, lines 45-67 – State the following: "From the perspective of memory, the data on the data bus at time slots 10, 11 are treated as write data. However, from the perspective of the requesting processor's address controller, such data is regarded as read data. This dichotomy, or "fork", enables the same data to simultaneously be sent to two separate destination locations, the proper memory address whereat an invalid version of the data has existed, and to the requesting CPU. The data recipient essentially neither knows nor cares from where the data having the desired DataID-SourceID match is coming. Thus, the ReadToShareFork implements a mechanism by which a singe data source coupled to the bus system can simultaneously update a memory location with valid data and also satisfy a read request. This simultaneous*

procedure is implemented by the expedient of putting onto the Address bus a duplicate of the original request or request ID, whereupon the original requestor can subsequently pull the now valid data. This procedure can be implemented with other than the Ignore signal mechanism. Further, an address controller could be caused to immediately have a ReadToShareFork placed on the bus. This procedure might arise if the address controller knew that memory update was necessary and wished to attain an earlier response in the system-wide sequence of transaction orders")

Claim 4. The apparatus of claim 1 and wherein the controller is further responsive to the invalid data signal to cause intervening data to be cleared, intervening data being data which was read from device locations other than that device location which produced the invalid data and which was read between the time data was read from that device location and the time data was read again from that device location, and then, after causing the data which was read again from the device to be sent to the desired destination, to cause the data to be read again from device locations which were read between the time data was read from that device location and the time data was read again from that device location and, subsequently, to send to the desired destination a data available signal for the data which was read again from those device locations. *(Column 33, lines 39-52 – State the following: "Deadlock can result due to same class non-cacheable and cacheable reads. More specifically, deadlock can occur if the following occur: a first UPA device first issues a non-cacheable read request (NCRD, NCBRD), and then issues a cacheable read request (RDS, RDSA, RDO or RDD), and both requests are outstanding, and the non-cacheable read request and the cacheable read request are in the same class, and the non-cacheable read request is directed to an IO bus (e.g. SBus) on a second UPA device, and a master on the same IO bus has issued a DMA cacheable read request to a location that is owned in the*

first UPA device. This request appears within the present invention after the cacheable read request from the first UPA, and the IO Bus will not issue a retry to the DMA cacheable read on the IO bus")

Claim 5. The apparatus of claim 1 and wherein the transaction queue further includes first pointer for pending transactions and a second pointer for dispatched transactions, and the controller is further responsive to the invalid data signal to cause the second pointer to assume the value of the first pointer. *(Column 38, lines 15-31 – Claim tags (a.k.a. pointers) for various transactions and the second tag assumes the value of the first)*

Claim 6. The apparatus of claim 1 wherein invalid data is data which has an uncorrectable error. *(Column 22, lines 27-28 – Describe invalid data as "fatal errors")*

Claim 7. An apparatus, comprising:

- a memory device to read and write data in response to device transactions, *(Column 2, lines 34-36 – State that a CPU will read or write data)*
- the device transactions including device read transactions and device write transactions; *(Column 8, lines 27-37 – Describe read and write transactions)*
- a data utilization device to accept data in response to a data available signal; *(Column 30, lines 53-63 – State the following: "The Local ReadToOwn transaction involves consideration of two cases. In one case, the device does not have a valid copy of the data, a case treated like other local reads, described above. Address Controller 180 waits for the data to be available in data input buffer ("DIB") 187, issues the S.sub.-- RBU S.sub.-- REPW, and then provides the data. In the second case, the device has a valid copy of the data. Here, Address Controller 180 issues an S.sub.-- OAK S.sub.-- REPLY to the device without waiting for the data. Because the Address Controller asserted*

Shared, neither memory nor another cache owning the data will respond with the data")

- a memory controller, coupled to the memory device and to the data utilization device, *(Column 7, lines 47-62 – Show that the memory controller, memory device, and data input and output buffers are coupled together)*
- to accept data from the memory device, to check the data for an error, to provide an invalid data signal if the data has an error, and to provide the data from the memory to the data utilization device; *(Column 25, lines 1-12 – State that there are "two error types, thus conforming to UPA interconnect architecture. More specifically, an invalid Data Bus packet can result from Read transactions for lines owned in a cache if the memory begins its response before the Owned signal is available. This condition may result, for example, in an attempt to reduce memory latency. Alternatively, if a write-type operation is "pushed" by the initiator, the destination cannot accept the data. The destination then cancels the Data Bus packet and assumes the responsibility to "pull" the data, a procedure described later herein with respect to WriteStream, WriteIO, and WriteBlock IO operations)*
- a transaction queue, coupled to the memory controller, to store pending device transactions and dispatched device transactions, *(Column 30, lines 24-30 – State that there is an input queue that stores transactions)*
- the queue including an input for receiving transactions, the queue including an output for dispatching transactions to the memory controller; and *(Column 30, lines 36-52 – Describe the queue as being able to input and output transactions)*
- a master controller, coupled to the memory controller, the transaction queue and the data utilization device, (*Column 7, lines 60-62 – State*

that the address controller contains an arbitration unit and memory controller)

- and responsive to an invalid data signal for a device read transaction to prevent the transaction queue from dispatching a pending transaction to the memory controller, to cause the transaction queue to dispatch again the device read transaction which resulted in the invalid data signal and, subsequently, to send to the data utilization device a data available signal for the data which resulted from the device read transaction which was dispatched again, and to enable the transaction queue to dispatch a pending transaction to the memory controller. *(Column 19, lines 4-24 – State the following: "Preferably, the ReadToShareFork can simultaneously (a) write valid data to a requested memory location containing invalid data, and (b) return the valid data to the requestor whose request for the data is still outstanding. Thus, rather than completing the outstanding request in approximately twenty clock cycles, the request is granted within about ten clock cycles at the same time that the data is updated in the relevant requested memory location. Note that from the standpoint of a memory controller the ReadToShareFork appears to implement a write transaction. However, from the standpoint of an address controller the ReadToShareFork appears to be implementing a read transaction")*

Claim 8. The apparatus of claim 7 wherein the master controller is further responsive to the invalid data signal to determine if invalid data read from the device location has occurred previously and, if not, to then cause the transaction queue to dispatch again the device read transaction which resulted in the invalid data signal and, subsequently, to send to the data utilization device a data available signal for the data which resulted from the device read transaction which was dispatched again, and to enable the transaction queue to dispatch a pending transaction to the memory

controller. *(Column 23, lines 51-67 and Column 24, lines 1-3 – State the following: "An asserted IGNORE signal blocks all address controllers from adding an IGNORE-flagged transaction into each address controller's coherent input queue. Thus, not even the address controller associated with a requesting device can load an IGNORE-flagged transaction into its own coherent input queue. The IGNORE-flagged transaction does not become asserted upon the omnibus system 20", and thus in a temporal or timing sense, the transaction has not yet occurred. (The transaction will be reissued by the device that asserted the IGNORE signal.) In essence, all address controllers 180 coupled to omnibus system 20" function as though the subject transaction had not occurred. The subject memory address (in which an invalid version of the desired data is retained) may still be accessed, but to no practical purpose since another memory location (which may be anywhere physically) holds the valid version of the desired data. When any CPU or other device attempts to access a cache line associated with an invalid state, as signified by associated memory TAG (or "MTAG"), global network interface 15 will cause the IGNORE signal to be asserted")*

Claim 9. The apparatus of claim 7 wherein the master controller is further responsive to the invalid data signal to determine if invalid data read from the device location has occurred previously and, if so, then to send to the data utilization device a data available signal for the data which resulted from the device read transaction which resulted in the invalid data signal. *(Column 19, lines 45-67 – State the following: "From the perspective of memory, the data on the data bus at time slots 10, 11 are treated as write data. However, from the perspective of the requesting processor's address controller, such data is regarded as read data. This dichotomy, or "fork", enables the same data to simultaneously be sent to two separate destination locations, the proper memory address whereat anbinvalid version of the data has existed, and to the requesting CPU. The data*

recipient essentially neither knows nor cares from where the data having the desired DataID-SourceID match is coming. Thus, the ReadToShareFork implements a mechanism by which a singe data source coupled to the bus system can simultaneously update a memory location with valid data and also satisfy a read request. This simultaneous procedure is implemented by the expedient of putting onto the Address bus a duplicate of the original request or request ID, whereupon the original requestor can subsequently pull the now valid data. This procedure can be implemented with other than the Ignore signal mechanism. Further, an address controller could be caused to immediately have a ReadToShareFork placed on the bus. This procedure might arise if the address controller knew that memory update was necessary and wished to attain an earlier response in the system-wide sequence of transaction orders")

Claim 10. The apparatus of claim 7 and wherein the master controller is further responsive to the invalid data signal to cause intervening data to be cleared, intervening data being data which was read from device locations other than that device location which produced the invalid data and which was read between the time data was read from that device location and the time data was read again from that device location, and then, after sending to the data utilization device a data available signal for the data which was read again from that device location, to cause the data to be read again from device locations which were read between the time data was read from that device location and the time data was read again from that device location and, subsequently, to send to the data utilization device a data available signal for the data which was read again from those device locations. *(Column 33, lines 39-52 – State the following: "Deadlock can result due to same class non-cacheable and cacheable reads. More specifically, deadlock can occur if the following occur: a first UPA device first issues a non-cacheable read request (NCRD, NCBRD), and then*

issues a cacheable read request (RDS, RDSA, RDO or RDD), and both requests are outstanding, and the non-cacheable read request and the cacheable read request are in the same class, and the non-cacheable read request is directed to an IO bus (e.g. SBus) on a second UPA device, and a master on the same IO bus has issued a DMA cacheable read request to a location that is owned in the first UPA device. This request appears within the present invention after the cacheable read request from the first UPA, and the IO Bus will not issue a retry to the DMA cacheable read on the IO bus")

Claim 11. The apparatus of claim 7 and wherein the transaction queue further includes first pointer for pending transactions and a second pointer for dispatched transactions, and the master controller is further responsive to the invalid data signal to cause the second pointer to assume the value of the first pointer. *(Column 38, lines 15-31 – Claim tags (a.k.a. pointers) for various transactions and the second tag assumes the value of the first)*

Claim 12. The apparatus of claim 7 wherein the memory controller, after checking the data for an error, is further operative to attempt to correct the invalid data and, if the invalid data cannot be corrected, then to send the invalid data signal. *(Column 1, lines 53-59 – State the following: "In write back cache, the data in main memory is "state" in that it is no longer correct, and only the cache holds a correct copy of the memory location. The modified copy of data in the cache is called "dirty". When the dirty data must be removed from the cache (to make room for a copy of a different memory location), the dirty data must be written back to memory")*

Claim 13. A method, comprising:

- receiving an indication that data read from a device location is not valid; *(Column 25, lines 3-6)*
- inhibiting write operations to the device; *(Column 24, lines 65-67 – State that the write operation cannot accept the data)*

- reading the data from that device location again; and enabling write operations to the device and sending to a desired destination a valid data signal for the data which resulted from reading that device location again. *(Column 19, lines 9-14 – State the following: "Writing to a memory location is relatively time consuming, e.g., taking perhaps 10 clock cycles. In the prior art, valid data might be written to the specific memory location after which the requester wishing to read the data would be permitted to do so. However obtaining the data and storing it locally would take yet another 10 cycles or so")*

Claim 14. The method of claim 13 and, prior to inhibiting write operations, if invalid data from that device location has not occurred previously *(Column 23, lines 51-67 and Column 24, lines 1-3 – State the following: "An asserted IGNORE signal blocks all address controllers from adding an IGNORE-flagged transaction into each address controller's coherent input queue. Thus, not even the address controller associated with a requesting device can load an IGNORE-flagged transaction into its own coherent input queue. The IGNORE-flagged transaction does not become asserted upon the omnibus system 20", and thus in a temporal or timing sense, the transaction has not yet occurred. (The transaction will be reissued by the device that asserted the IGNORE signal.) In essence, all address controllers 180 coupled to omnibus system 20" function as though the subject transaction had not occurred. The subject memory address (in which an invalid version of the desired data is retained) may still be accessed, but to no practical purpose since another memory location (which may be anywhere physically) holds the valid version of the desired data. When any CPU or other device attempts to access a cache line associated with an invalid state, as signified by associated memory TAG (or "MTAG"), global network interface 15 will cause the IGNORE signal to be asserted")*

- then proceeding with inhibiting write operations, *(Column 24, lines 65-67 – State that the write operation cannot accept the data)*
- reading the data again, enabling write operations and sending the valid data signal. *(Column 19, lines 9-14 – State the following: "Writing to a memory location is relatively time consuming, e.g., taking perhaps 10 clock cycles. In the prior art, valid data might be written to the specific memory location after which the requester wishing to read the data would be permitted to do so. However obtaining the data and storing it locally would take yet another 10 cycles or so")*

Claim 16. The method of claim 13 in which invalid data is data which has an uncorrectable error. *(Column 22, lines 27-28 – Describe invalid data as "fatal errors")*

Claim 17. A method, comprising:

- receiving an indication that data read from a device location is invalid; *(Column 25, lines 3-6)*
- if invalid data read from the device location has occurred previously then sending a valid data signal for the data to a desired destination; *(Column 19, lines 45-67 – State the following: "From the perspective of memory, the data on the data bus at time slots 10, 11 are treated as write data. However, from the perspective of the requesting processor's address controller, such data is regarded as read data. This dichotomy, or "fork", enables the same data to simultaneously be sent to two separate destination locations, the proper memory address whereat an invalid version of the data has existed, and to the requesting CPU. The data recipient essentially neither knows nor cares from where the data having the desired DataID-SourceID match is coming. Thus, the ReadToShareFork implements a mechanism by which a singe data source coupled to the bus system can simultaneously update a memory location with valid data and also satisfy a read request. This simultaneous procedure is implemented by*

the expedient of putting onto the Address bus a duplicate of the original request or request ID, whereupon the original requestor can subsequently pull the now valid data. This procedure can be implemented with other than the Ignore signal mechanism. Further, an address controller could be caused to immediately have a ReadToShareFork placed on the bus. This procedure might arise if the address controller knew that memory update was necessary and wished to attain an earlier response in the system-wide sequence of transaction orders")

- and if invalid data read from the device location has not occurred previously then: *(Column 23, lines 51-67 and Column 24, lines 1-3 – State the following: "An asserted IGNORE signal blocks all address controllers from adding an IGNORE-flagged transaction into each address controller's coherent input queue. Thus, not even the address controller associated with a requesting device can load an IGNORE-flagged transaction into its own coherent input queue. The IGNORE-flagged transaction does not become asserted upon the omnibus system 20", and thus in a temporal or timing sense, the transaction has not yet occurred. (The transaction will be reissued by the device that asserted the IGNORE signal.) In essence, all address controllers 180 coupled to omnibus system 20" function as though the subject transaction had not occurred. The subject memory address (in which an invalid version of the desired data is retained) may still be accessed, but to no practical purpose since another memory location (which may be anywhere physically) holds the valid version of the desired data. When any CPU or other device attempts to access a cache line associated with an invalid state, as signified by associated memory TAG (or "MTAG"), global network interface 15 will cause the IGNORE signal to be asserted")*

- inhibiting write operations to the device; *(Column 24, lines 65-67 – State that the write operation cannot accept the data)*
- reading the data in that device location again; and enabling write operations to the device and sending to a desired destination a valid data signal for the data which resulted from reading that device location again. *(Column 19, lines 9-14 – State the following: "Writing to a memory location is relatively time consuming, e.g., taking perhaps 10 clock cycles. In the prior art, valid data might be written to the specific memory location after which the requester wishing to read the data would be permitted to do so. However obtaining the data and storing it locally would take yet another 10 cycles or so")*

Claim 18. The method of claim 17 and, prior to inhibiting write operations, if invalid data from that device location has not occurred previously *(Column 23, lines 51-67 and Column 24, lines 1-3 – State the following: "An asserted IGNORE signal blocks all address controllers from adding an IGNORE-flagged transaction into each address controller's coherent input queue. Thus, not even the address controller associated with a requesting device can load an IGNORE-flagged transaction into its own coherent input queue. The IGNORE-flagged transaction does not become asserted upon the omnibus system 20", and thus in a temporal or timing sense, the transaction has not yet occurred. (The transaction will be reissued by the device that asserted the IGNORE signal.) In essence, all address controllers 180 coupled to omnibus system 20" function as though the subject transaction had not occurred. The subject memory address (in which an invalid version of the desired data is retained) may still be accessed, but to no practical purpose since another memory location (which may be anywhere physically) holds the valid version of the desired data. When any CPU or other device attempts to access a cache line associated with an invalid state, as signified by associated memory TAG*

(or "MTAG"), global network interface 15 will cause the IGNORE signal to be asserted")

- then proceeding with inhibiting write operations, *(Column 24, lines 65-67 – State that the write operation cannot accept the data)*
- reading the data again, enabling write operations and sending the valid data signal. *(Column 19, lines 9-14 – State the following: "Writing to a memory location is relatively time consuming, e.g., taking perhaps 10 clock cycles. In the prior art, valid data might be written to the specific memory location after which the requester wishing to read the data would be permitted to do so. However obtaining the data and storing it locally would take yet another 10 cycles or so")*

Claim 20. The method of claim 17 in which invalid data is data which has an uncorrectable error. *(Column 22, lines 27-28 – Describe invalid data as "fatal errors")*

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 15 and 19 are rejected under 35 U.S.C.103(a) as being unpatentable over Hagersten et al. as applied to claims 13 and 17 above, further in view of Brzezinski et al. (US Patent 5,570,297).

Hagersten teaches the limitations of claims 13 and 17 for the reasons above.

Hagersten 's invention differs from the claimed invention in that there is no specific reference to error flags.

Hagersten fails to teach claims 15 and 19, which state: "if invalid data read from the device location has not occurred previously then setting a previous error flag to indicate that invalid data read from that device location has occurred previously; and if invalid data read from the device location has occurred previously then clearing the previous error flag to indicate that invalid data read from that device location has not occurred previously." However, Brzezinski 's invention discloses the following: "If the signal is a start bit, the next successive eight bits are read as data and the next successive bit is checked at 66 to determine whether it is a stop bit. If not, an error flag is set at 68 (or the error flag is cleared at 70), and the subroutine provides an output at 72 indicating that data is available. The presence of the error flag serves to indicate that an error has occurred due to some failure such as not receiving a stop bit or other indication that invalid data is being received." (Column 7, lines 41-49). It would have been obvious to one of ordinary skill in the art, having the teachings of the "Optimizing Responses in a Coherent Distributed Electronic System" of Hagersten and Brzezinski 's "Method and Apparatus for Synchronizing Data Transfer Rate From a Cathode Ray Tube Video Monitor to a Portable Information Device" before him at the time the invention was made, to include error flags so that errors and invalidated data would be easily and visibly noted, which would increase overall efficiency in the system.

Response to Arguments

7. Applicant's arguments (filed April 25, 2006) with respect to claims 1-20 have been considered but are moot in view of the previous, new and following ground(s) of rejection.

8. With regards to Claim 1, the Applicant alleges that "Hagerston does not teach an apparatus having a transaction queue adapted to: being prevented from dispatching a pending

transaction to a device, dispatching again a device read transaction to a memory location which resulted in an invalid data signal and, subsequently, dispatching a pending transaction to the device". The above statement will be traversed in parallel with the Hagerston reference in Column 19, lines 4-24: an apparatus having a transaction queue adapted to: being prevented from dispatching a pending transaction to a device *(Column 35, lines 12-15 – State the following: In further contrast, transactions that interact with pending transactions need not be blocked. Instead, interface 30 deals with such interaction using simple pipeline bypasses in the tags pipeline)*, dispatching again a device read transaction to a memory location which resulted in an invalid data signal and, subsequently, dispatching a pending transaction to the device *(Column 19, lines 4-24 – State the following: "Preferably, the ReadToShareFork can simultaneously (a) write valid data to a requested memory location containing invalid data, and (b) return the valid data to the requestor whose request for the data is still outstanding. Thus, rather than completing the outstanding request in approximately twenty clock cycles, the request is granted within about ten clock cycles at the same time that the data is updated in the relevant requested memory location. Note that from the standpoint of a memory controller the ReadToShareFork appears to implement a write transaction. However, from the standpoint of an address controller the ReadToShareFork appears to be implementing a read transaction)*. Therefore, the Applicant's arguments are moot in view of the prior art.

9. The Applicant further alleges that Hagerston does not teach "reading a memory device location multiple times in an attempt to correctly read data which may have become unstable or corrupt due to a transient such that a second or successive data read may result in correct data being read". However, the Applicant's claims also do not disclose "reading a memory device

location multiple times", so the Applicant's arguments are irrelevant and remain moot in view of the prior art.

10. The Applicant alleges that "the RTSF transaction can read and write at the same time", but the Applicant claims no such feature. This argument is irrelevant, due to the prior art may very well have extra components not necessarily in the application. Therefore, the Applicant's argument is moot.

11. The Applicant further alleges that Hagerston does not teach "rereading a memory location providing invalid data, and then subsequently, sending to a desired destibation a data available signal for the data which resulted from data reread, and dispatch the pending transaction". However, the Applicant's claims also do not disclose "rereading a memory location providing invalid data", so the Applicant's arguments are irrelevant and remain moot in view of the prior art.

12. The Applicant further alleges that Hagerston does not "disclose inhibition of write operations". However, the Examiner respectfully points to the following in Hagerston: "The ReadToOwn transaction is initiated by a board wanting exclusive ownership of the cache line so that it may write to it. The same transaction is used regardless of whether the initiator has a valid cached copy of the line. With respect to snooping and the ReadToOwn transaction, the board that has memory corresponding to this address asserts Mapped. Otherwise, if Mapped is not asserted, the initiator will not expect a response and will return an error to the device. All boards snoops on the Address Bus packet, and a board owning the line asserts Owned and will respond to the initiator. There can be at most one owner for the line. If the initiator already has a valid copy of the line, it may not need the data but uses the ReadToOwn transaction to obtain

exclusive ownership of the line. In this case, the initiator may also assert Shared to inhibit the response from another cache or Memory that may own the Line" (Column 14, lines 59-67 and Column 15, lines 1-8). Since the above clearly teaches what the Applicant argued against, the Applicant's arguments are moot in view of the prior art.

13. With regards to Claims 6 and 16, the Applicant alleges that Hagerston does not discuss the "data having an uncorrectable error". However, by definition, a fatal error (as pointed out in the original rejection), is one that is uncorrectable. Therefore, the Applicant's argument is moot in view of Hagerston.

14. With regards to Claims 15 and 19, the Applicant alleges that "those skiled in the art...would not turn to Hagerston or Brzezinski for guidance, since both references are directed t solving problems not pertinent to the problems solved by the Applicant's currently claimed invention". Whether or not the prior art is geared towards solving the same problems as the claimed invention is irrelevant. The concept of combining Hagerston and Brzezinski allows for a single invention that has the potential of solving the problems proposed by Claims 15 and 19. Therefore, the Applicant's argument is moot in view of Hagerston and Brzezinski.

15. The Applicant alleges that "Brzezinski discloses to test for a stop bit", but the Applicant claims no such feature. This argument is irrelevant, due to the prior art may very well have extra components not necessarily in the application. Therefore, the Applicant's argument is moot.

16. Further in view of Claims 15 and 19, the Applicant alleges that Brzezinski teaches the opposite of what the claims teach. Since this is a 103(a) rejection, it is obvious to combine items in different sequences to reach similar goals. Furthermore, Brzezinski's flag clearly indicates

whether or not invalid data has occurred, as shown in the rejections above. Therefore, the Applicant's argument is moot in view of Hagerston and Brzezinski.

17. In view of the responses above, the Examiner therefore did in fact originally set forth a prima facie case of obviousness.

18. Any claims not mentioned in the Examiner's response to Applicant's arguments stand rejected due to their dependence on the rejected independent claims.

Conclusion

19. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

20. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lev I. Iwashko whose telephone number is (571)272-1658. The examiner can normally be reached on M-Th, from 8-6PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matt Kim can be reached on (571)272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Lev Iwashko

Lev Iwashko

SCElmore

STEPHEN C. ELMORE
PRIMARY EXAMINER